

## 8. feladat

Írj olyan programot, amely egy a felhasználó által a penz.txt fájlban elhelyezett összegre megmondja, hogy hogy lehet a legkevesebb érmevel kifizetni. Feltesszük, hogy 1, 2, 5, 10, 20, 50, 100 és 200 Ft-os érmeink vannak. A kiírásnál azok a pénzérmék ne szerepeljenek, amikből 0 db kellene.

Példa:

```
// a program beolvassa a penz.txt tartalmát, vagyis egy darab egész számot
```

```
// tegyük föl, hogy a szám 1737
```

```
// a program kikalkulálja, hogy lehet a legkevesebb érmevel kifizetni
```

```
    8 db 200Ft-os
```

```
    1 db 100Ft-os
```

```
    1 db 20Ft-os
```

```
    1 db 10Ft-os
```

```
    1 db 5Ft-os
```

```
    1 db 2Ft-os
```

A program a példában szereplő kézzel megjelenített sorokat nem írja ki. Figyeljünk oda arra is, hogy nincs 5-ös kerekítés, mert a példában használunk 1 és 2 Ft-os érmeket is.

## 8. feladat

```
/*8_házi*/
# include<stdio.h>
# include<fstream.h>
int main()
{int i,db,osszeg,cimlet[8]={ 200,100,50,20,10,5,2,1 };
//Fájl tartalmának beolvasása
FILE *fp1;
fp1=fopen("penz.txt","rt");
fscanf(fp1,"%d",&osszeg);
fclose(fp1);
//Címletek számának meghatározása
for(i=0;i<=7;i++)
{db=osszeg/cimlet[i];osszeg=osszeg-db*cimlet[i];
if (db>0) {printf("\n%d db %d Ft-os \n",db,cimlet[i]);db=0;}
}
printf("\n");
system("pause");
return 0;
} //main
```

A címletek megadása tömbbe.  
Vigyázni kell, mert a cimlet[0]-ba teszi az első címletet (200)!

'penz.txt' megnyitása  
olvasásra.

Címletekkel való műveletek. 0-tól indul a  
ciklus, mert a 200 a cimlet[0]-ban van.

A hányados felveszi a 'db' deklarációját  
(egész). Ez tehát egy egész osztás!

Létező címletek kiírása.